10-26-00 A

Please type a plus sign (+) inside this box → [+]

PTO/SB/05 (12/97)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

JC953 U.S. PTO 10/25/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	TIF-29339
First Named Inventor or Application Identifier	Gerard Chauvel, Et. Al.
Title	Intelligent Power Management For Distributed Processing Systems
Express Mail Label No.	EL360238921US

JC920 U.S. PTO 09/696052 10/25/00

APPLICATION ELEMENTS

See MPEP Chapter 600 concerning utility patent application contents

1. [X] *Fee Transmittal Form *(e.g., PTO/SB/17)* *(Submit an original, and a duplicate for fee processing)*
2. [X] Specification *[Total Pages* [19] *]* *(preferred arrangement set forth below)*
   - Descriptive title of the Invention
   - Cross References to Related Applications
   - Statement Regarding Fed sponsored R&D
   - Reference to Microfiche Appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
3. [X] Drawing(s) *(35 USC d113)* *[Total Sheets* [8] *]*
4. Oath or Declaration *[Total Pages* [ ] *]*
   - a. [ ] Newly Executed (original or copy)
   - b. [ ] Copy from a prior application (37 CFR §1.63(d)) *(for continuation/divisional with Box 17 completed)*

   [Note Box 5 below]

   - i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR §1.63(d)(2) and 1.33(b).
5. [ ] Incorporation By Reference *(useable if Box 4b is checked)* The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program *(Appendix)*
7. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   - a. [ ] Computer Readable Copy
   - b. [ ] Paper Copy (identical to computer copy)
   - c. [ ] Statement verifying identical of above copies

ADDRESS TO:

Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

ACCOMPANYING APPLICATION PARTS

8. [ ] Assignment Papers (cover sheet & Documents(s))
9. [ ] 37 CFR §3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
12. [X] Preliminary Amendment
13. [X] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
14. [ ] *Small Entity Statement(s) *(PTO/SB/09-12)* [ ] Statement filed in prior application Status still proper and desired
15. [ ] Certified Copy of Priority Document(s) *if foreign priority is claimed)*
16. [ ] Other:

*A new statement is required to be entitled to pay small entity fees, except where one has been filed in a prior application and is being relied upon

17. **If a CONTINUING APPLICATION,** *check appropriate box and supply the requisite information below and in a preliminary amendment:*

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: /

Prior application information: ***Examiner*** ***Group / Art Unit:***

18. CORRESPONDENCE ADDRESS

[ ] *Customer Number or Bar Code Label* *(Insert Customer No. or Attach bar code label here)* or [X] *Correspondence address below*

NAME	RONALD O. NEERINGS Texas Instruments Incorporated				
ADDRESS	Mail Station 3999 P. O. Box 655474				
CITY	Dallas	STATE	TX	ZIP CODE	75265
COUNTRY	U.S.A.	TELEPHONE	(972) 917-5299	FAX	(972) 917-4418

Name (Print/Type)	RONALD O. NEERINGS	Registration No. (Attorney/Agent)	34,227
Signature	Ronald O. Neerings	Date	10/25/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231

PTO/SB/17 (1/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL

Patent fees are subject to annual revision on October 1.
These are the fees effective October 1, 1997
Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12.

TOTAL AMOUNT OF PAYMENT	($) 746.00

Complete If Known	
Application Number	TBD
Filing Date	October 25, 2000
First Named Inventor	Gerard Chauvel, Et. al.
Examiner Name	TBE
Group / Art Unit	TBD
Attorney Docket No.	TIE-29339

JC920 U.S. PTO 09/696052 10/25/00

METHOD OF PAYMENT

1. ☒ The Commissioner is hereby authorized to charge to the following Deposit Account,

Deposit Account Number: 20-0668

Deposit Account Name: Texas Instruments Incorporated

☐ Charge any additional fee required or credit any overpayment

☒ Charge all indicated fees and any additional fee required or credit any overpayment

2. ☐ **Payment Enclosed:**

☐ Check ☐ Money Order ☐ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Fee Code	Entity Fee ($)	Small Fee Code	Entity Fee ($)	Fee Description	Fee Paid
101	790	201	395	Utility filing fee	$710
106	330	206	165	Design filing fee	$
107	540	207	270	Plant filing fee	$
108	790	208	395	Reissue filing fee	$
114	150	214	75	Provisional filing fee	$

SUBTOTAL (1) ($)710

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	22	-20**=	2	x	18	=	36
Independent Claims	2	-3** =	0	x	78	=	0
Multiple Dependent					0	=	0

***or number previously paid, if greater; For Reissue, see below*

Large Fee Code	Entity Fee ($)	Small Fee Code	Entity Fee ($)	Fee Description
103	22	203	11	Claims in excess of 20
102	82	202	41	Independent Claims in excess of 3
104	270	204	135	Multiple dependent claims in excess of 3
109	82	209	41	**Reissue independent claims over original patent
110	22	210	11	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)36

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Fee Code	Entity Fee ($)	Small Fee Code	Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension of time within second month	
117	950	217	475	Extension of time within third month	
118	1,510	218	755	Extension of time within fourth month	
128	2,060	228	1,030	Extension of time within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a pubic use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,320	241	660	Petition to revive - unintentional	
142	1,320	242	660	Utility issue fee (or reissue)	
143	450	243	225	Design issue fee	
144	670	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt.	
581	40	581	40	Recording each patent assignment per properly (time number of properties)	
146	790	246	395	Filing a submission after final rejection (37 CFR 1.129(a))	
149	790	249	395	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) _____

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)

SUBMITTED BY				Complete (if applicable)	
Typed or Printed Name	Ronald O. Neerings			Reg. Number	34,227
Signature	Ronald O. Neerings	Date	10/25/00	Deposit Account User ID	

Burden Hour Statement. This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231 DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO: Assistant Commissioner for Patents, Washington, DC 20231

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Gerard Chauvel, et al.	Art Unit :	TBD
Serial No.:	TBD	Examiner:	TBD
Filed:	10/25/2000	Docket:	TIF-29339
For:	Intelligent Power Management For Distributed Processing Systems		

PRELIMINARY AMENDMENT

October 25, 2000

MAILING CERTIFICATE UNDER 37 C.F.R. §1.8(A)

I hereby certify that on or before this date the above correspondence is being deposited with the U.S. Postal Service as First Class Mail in an envelope addressed to: Assistant Commissioner For Patents, Washington, D.C. 20231.

Elizabeth Austin — Date 10/25/2000

Assistant Commissioner For Patents

Washington, D. C. 20231

Sir:

Please amend the above-referenced application as follows:

In the Specification:

On page 1, after the title and before the first line of text, insert a new paragraph --This application claims priority under 35 U.S.C. §119(e)(1) of European Patent Application Number 99402655.7, filed October 25, 1999, the entirety of which is incorporated herein by reference.

Respectfully submitted,

Ronald O. Neerings
Attorney for Applicant
Reg. No.: 34,227

Texas Instruments Incorporated
P. O. Box 655474, M/S 3999
Dallasl, Texas 75265
(972) 917-5299

INTELLIGENT POWER MANAGEMENT FOR DISTRIBUTED PROCESSING SYSTEMS

CROSS-REFERENCE TO RELATED APPLICATIONS

Not Applicable

STATEMENT OF FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable

BACKGROUND OF THE INVENTION

1. TECHNICAL FIELD

This invention relates in general to integrated circuits and, more particularly, to managing power in a processor.

2. DESCRIPTION OF THE RELATED ART

For many years, the focus of processor design, including designs for microprocessor units (MPUs), co-processors and digital signal processors (DSPs), has been to increase the speed and functionality of the processor. Presently, power consumption has become a serious issue. Importantly, maintaining low power consumption, without seriously impairing speed and functionality, has moved to the forefront in many designs. Power consumption has become important in many applications because many systems, such as smart phones, cellular phones, PDAs (personal digital assistants), and handheld computers

operate from a relatively small battery. It is desirable to maximize the battery life in these systems, since it is inconvenient to recharge the batteries after short intervals.

Currently, approaches to minimizing power consumption involve static power management; i.e., designing circuits which use less power. In some cases, dynamic actions have been taken, such as reducing clock speeds or disabling circuitry during idle periods.

While these changes have been important, it is necessary to continuously improve power management, especially in systems where size and, hence, battery size, is important to the convenience of using a device.

In addition to overall power savings, in a complex processing environment, the ability to dissipate heat from the integrated circuit becomes a factor. An integrated circuit will be designed to dissipate a certain amount of heat. If tasks require multiple systems on the integrated circuit to draw high levels of current, it is possible that the circuit will overheat, causing system failure.

In the future, applications executed by integrated circuits will be more complex and will likely involve multiprocessing by multiple processors, including MPUs, DSPs, coprocessors and DMA channels in a single integrated circuit (hereinafter, a "distributed processing system"). DSPs will evolve to support multiple, concurrent applications, some of which will not be dedicated to a specific DSP platform, but will be loaded from a global network such as the Internet. Accordingly, the tasks that a distributed processing system will be able to handle without overheating will become uncertain.

Accordingly, a need has arisen for a method and apparatus for managing power in a circuit without seriously impacting performance.

BRIEF SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for controlling the execution of tasks in a processor comprising a plurality of processing modules. Consumption information is calculated based on probabilistic values for activities associated with the tasks. Tasks are then executed on the processing modules responsive to the consumption information.

The present invention provides significant advantages over the prior art. First, it provides for a fully dynamic power management. As the tasks executed in the processing system change, the power management software can build new scenarios to ensure that thresholds are not exceeded. Further, as environmental conditions change, such as battery voltages dropping, the power management software can re-evaluate conditions and change scenarios, if necessary. Second, the power management software is transparent to the various tasks that it controls. Thus, even if a particular task does not provide for any power management, the power management software assumes responsibility for executing the task in a manner that is consistent with the power capabilities of the processing system. Third, the overall operation of the power management software can be used with different hardware platforms, with different hardware and tasks accommodated by changing profiles used for making the power calculations.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

Figure 1 illustrates a block diagram of a distributed processing system;

Figure 2 illustrates a software layer diagram for the distributed processing system;

Figure 3 illustrates an example showing the advantages of power management for a distributed processing system;

Figures 4a and 4b illustrate flow diagrams showing preferred embodiments for the operation of the power management software of Figure 2;

Figure 5 illustrates the building system scenario block of Figure 4;

Figure 6 illustrates the activities estimate block of Figure 4;

Figure 7 illustrates the power compute block of Figure 4;

Figure 8 illustrates the activity measure and monitor block of Figure 4;

Figure 9 illustrates a block diagram showing the distributed processing system with activity counters.

DETAILED DESCRIPTION OF THE INVENTION

The present invention is best understood in relation to Figures 1-9 of the drawings, like numerals being used for like elements of the various drawings.

Figure 1 illustrates a general block diagram of a general distributed processing system 10, including an MPU 12, one or more DSPs 14 and one or more DMA channels or coprocessors (shown collectively as DMA/Coprocessor 16). In this embodiment, MPU 12 includes a core 18 and a cache 20. The DSP 14 includes a processing core 22 and a local memory 24 (an actual embodiment could use separate instruction and data memories, or could use a unified instruction and data memory). A memory interface 26 couples a shared memory 28 to one or more of the MPU 12, DSP 14 or DMA/Coprocessor 16. Each processor (MPU 12, DSPs 14) can operate in full autonomy under its own operating system (OS) or real-time operating system (RTOS) in a real distributed processing system, or the MPU 12 can operate the global OS that supervises shared resources and memory environment.

Figure 2 illustrates a software layer diagram for the distributed processing system 10. As shown in Figure 1, the MPU 12 executes the OS, while the DSP 14 executes an RTOS. The OS and RTOSs comprise the OS layer 30 of the software. A distributed application layer 32 includes JAVA, C++ and other applications 34, power management tasks 38 which use profiling data 36 and a global tasks scheduler 40. A middleware software layer 42 communicates between the OS layer 30 and the applications in the distributed application layer 32.

Referring to Figures 1 and 2, the operation of the distributed processing system 10 is discussed. The distributed processing system 10 can execute a variety of tasks. A typical application for the distributed processing system 10 would be in a smartphone application where the distributed processing system 10 handles wireless communication, video and audio decompression, and user

interface (i.e., LCD update, keyboard decode). In this application, the different embedded systems in the distributed processing system 10 would be executing multiple tasks of different priorities. Typically, the OS would perform the task scheduling of different tasks to the various embedded systems.

The present invention integrates energy consumption as a criterion in scheduling tasks. In the preferred embodiment, the power management application 38 and profiles 36 from the distributed applications layer 32 are used to build a system scenario, based on probabilistic values, for executing a list of tasks. If the scenario does not meet predetermined criteria, for example if the power consumption is too high, a new scenario is generated. After an acceptable scenario is established, the OS layer monitors the hardware activity to verify that the activity predicted in the scenario was accurate.

The criteria for an acceptable task scheduling scenario could vary depending upon the nature of the device. One important criterion for mobile devices is minimum energy consumption. As stated above, as electronic communication devices are further miniaturized, the smaller battery allocation places a premium on energy consumption. In many cases during the operation of a device, a degraded operating mode for a task may be acceptable in order to reduce power, particularly as the batteries reach low levels. For example, reducing the LCD refresh rate will decrease power, albeit at the expense of picture quality. Another option is to reduce the MIPs (millions of instructions per second) of the distributed processing system 10 to reduce power, but at the cost of slower performance. The power management software 38 can analyze different scenarios using different combinations of degraded performance to reach acceptable operation of the device.

Another objective in managing power may be to find the highest MIPs, or lowest energy for a given power limit setup.

Figures 3a and 3b illustrate an example of using the power management application 38 to prevent the distributed processing system 10 from exceeding an average power dissipation limit. In Figure 3a, the DSP 14, DMA 16 and MPU 12 are concurrently running a number of tasks. At time t1, the average power dissipation of the three embedded systems exceeds the average limit imposed on the distributed processing system 10. Figure 3b illustrates a scenario where the same tasks are executed; however, an MPU task is delayed until after the DMA and DSP tasks are completed in order to maintain an acceptable average power dissipation profile.

Figure 4a illustrates a flow chart describing operation of a first embodiment of the power management tasks 38. In block 50, the power management tasks are invoked by the global scheduler 40, which could be executed on the MPU 12 or one of the DSPs 14; the scheduler evaluate the upcoming application and splits it into tasks with associated precedence and exclusion rules. The task list 52 could include, for example, audio/video decoding, display control, keyboard control, character recognition, and so on. In step 54, the task list 52 is evaluated in view of the task model file 56 and the accepted degradations file 58. The task model file 56 is part of the profiles 36 of the distributed applications layer 32. The task model file 56 is a previously generated file that assigns different models to each task in the task list. Each model is a collection of data, which could be derived experimentally or by computer aided software design techniques, which defines characteristics of the associated task, such as latency constraints, priority, data flows, initial energy estimate at a reference processor speed, impacts of degradations, and an execution profile on a given processor as a function of MIPs and time. The degradation list 58 sets forth the variety of degradations that can be used in generating the scenario.

Each time the task list is modified (i.e., a new task is created or a task is deleted) or when a real time event occur, based on the task list 52 and the task model 56 in step 54, a scenario is built. The scenario allocates the various tasks to the modules and provides priority information setting the priority with which tasks are executed. A scenario energy estimate 59 at a reference speed can be computed from the tasks' energy estimate. If necessary or desirable, tasks may be degraded; i.e., a mode of the task that uses fewer resources may be substituted for the full version of a task. From this scenario, an activities estimate is generated in block 60. The activities estimate uses task activity profiles 62 (from the profiling data 36 of the distributed application layer 32) and a hardware architectural model 64 (also from the profiling data 36 of the distributed application layer 32) to generate probabilistic values for hardware activities that will result from the scenario. The probabilistic values include each module's wait/run time share (effective MHz), accesses to caches and memories, I/O toggling rates and DMA flow requests and data volume. Using a period T that matches the thermal time constant, from the energy estimate 59 at a reference processor speed and the average activities derived in step 60 (particularly, effective processors speeds), it is possible to compute an average power dissipation that will be compared to thermal package model. If the power value exceeds any thresholds set forth in the package thermal model 72, the scenario is rejected in decision block 74. In this case, a new scenario is built in block 54 and steps 60, 66 and 70 are repeated. Otherwise, the scenario is used to execute the task list.

During operation of the tasks as defined by the scenario, the OS and RTOSs track activities by their respective modules in block 76 using counters 78 incorporated in the hardware. The actual activity in the modules of the distributed processing system 10 may vary from the activities estimated in block 60. The data from the hardware counters are monitored on a T periodic basis to

produce measured activity values. These measured activity values are used in block 66 to compute an energy value for this period, and hence, an average power value in block 66, as described above, and are compared to the package thermal model in block 72. If the measured values exceed thresholds, then a new scenario is built in block 54. By continuously monitoring the measured activity values, the scenarios can be modified dynamically to stay within predefined limits or to adjust to changing environmental conditions.

Total energy consumption over T for the chip is calculated as:

$$E = \int_T \sum_{\text{modules}} \left[\alpha \cdot Cpd \cdot f \cdot V_{dd}^2\right] \cdot dt \equiv \sum_{\text{modules}} \left[\sum_T (\alpha \cdot)\right] \cdot Cpd \cdot f \cdot V_{dd}^2$$

where, f is the frequency, V_{dd} is the supply voltage and α is the probabilistic (or measured, see discussion in connection with block 76 of this figure) activity. In other words, $\sum_T (\alpha) * Cpd * f * V_{dd}^2$ is the energy corresponding to a particular hardware module characterized by equivalent dissipation capacitance Cpd ; counters values give $\sum_T (\alpha)$ and E is the sum of all energies for all modules in the distributed processing system 10 dissipated within T. Average system power dissipation W = E/T. In the preferred embodiment, measured and probabilistic energy consumption is calculated and the average power dissipation is derived from the energy consumption over period T. In most cases, energy consumption information will be more readily available. However, it would also be possible to calculate the power dissipation from measured and probabilistic power consumption.

Figure 4b is a flow chart describing operation of a second embodiment of the power management tasks 38. The flow of Figure 4b is the same as that of Figure 41, except when the scenario construction algorithm is invoked (new task, task delete, real time event) in step 50, instead of choosing one new scenario, *n* different scenarios that match the performances constraints can be pre-computed

in advance and stored in steps 54 and 59, in order to reduce the number of operations within the dynamic loop and provide faster adaptation if the power computed in the tracking loop leads to current scenario rejection in block 74. In Figure 4b, if the scenario is rejected, another pre-computed scenario is selected in block 65. Otherwise the operation is the same as shown in Figure 4a.

Figures 5 – 8 illustrate the operation of various blocks of Figure 3 in greater detail. The build system block 54 is shown in Figure 5. In this block, a task list 52, a task model 56, and a list of possible task degradations 58 are used to generate a scenario. The task list is dependent upon which tasks are to be executed on the distributed processing system 10. In the example of Figure 5, three tasks are shown: MPEG4 decode, wireless modem data receive and keyboard event monitor. In an actual implementation, the tasks could come from any number of sources. The task model sets forth conditions which must be taken in consideration in defining the scenario, such as latency and priority constraints, data flow, initial energy estimates, and the impact of degradations. Other conditions could also be used in this block. The output of the build system scenario block is a scenario 80, which associates the various tasks with the modules and assigns priorities to each of the tasks. In the example shown in Figure 5, for example, the MPEG4 decode task has a priority of 16 and the wireless modem task has a priority of 4.

The scenarios built in block 54 could be based on a number of different considerations. For example, the scenarios could be built based on providing the maximum performance within the packages thermal constraints. Alternatively, the scenarios could be based on using the lowest possible energy. The optimum scenario could change during operation of a device; for example, with fully charged batteries a device may operate at a maximum performance level. As the

power in the batteries diminished below a preset level, the device could operate at the lowest possible power level to sustain operation.

The scenario 80 from block 54 is used by the activities estimate block 60, shown in Figure 6. This block performs a probabilities computation for various parameters that affect power usage in the distributed processing system 10. The probabilistic activities estimate is generated in conjunction with task activity profiles 62 and hardware architectural models 64. The task activity profiles include information on the data access types (load/store) and occurrences for the different memories, code profiles, such as the branches and loops used in the task, and the cycles per instruction for instructions in the task. The hardware architectural model 64 describes in some way the impact of the task activity profiles 62 on the system latencies, that will permit computation of estimated hardware activities (such as processor run/wait time share). This model takes into account the characteristics of the hardware on which the task will be implemented, for example, the sizes of the caches, the width of various buses, the number of I/O pins, whether the cache is write-through or write back, the types of memories used (dynamic, static, flash, and so on) and the clock speeds used in the module. Typically, the model can consist of a family of curves that represent MPU and DSP effective frequency variations with different parameters, such as data cacheable/non-cacheable, read/write access shares, number of cycles per instruction, and so on. In the illustrated embodiment of Figure 6, values for the effective frequency of each module, the number of memory accesses, the I/O toggling rates and the DMA flow are calculated. Other factors that affect power could also be calculated.

The power compute block 66 is shown in Figure 8. In this block, the probabilistic activities from block 60 or the measured activities from block 76 are used to compute various energy values and, hence, power values over a period

T. The power values are computed in association with hardware power profiles, which are specific to the hardware design of the distributed processing system 10. The hardware profiles could include a Cpd for each module, logic design style (D-type flip-flop, latches, gated clocks and so on), supply voltages and capacitive loads on the outputs. Power computations can be made for integrated modules, and also for external memory or other external devices.

Activity measure and monitor block 76 is shown in Figure 8. Counters are implemented throughout the distributed processing system 10 to measure activities on the various modules, such as cache misses, TLB (translation lookaside buffer) misses, non-cacheable memory accesses, wait time, read/write requests for different resources, memory overhead and temperature. The activity measure and monitor block 76 outputs values for the effective frequency of each module, the number of memory accesses, the I/O toggling rates and the DMA flow. In a particular implementation, other values may also be measured. The output of this block is sent to the power compute block 66.

Figure 9 illustrates and example of a distributed processing system 10 using power/energy management software. In this example, the distributed processing system 10 includes a MPU 12, executing an OS, and two DSPs 14 (individually referenced as DSP1 14a and DSP2 14b), each executing a respective RTOS. Each module is executing a monitor task 82, which monitors the values in various activity counters 78 throughout the distributed processing system 10. The power compute task is executed on DSP 14a. The various monitor tasks retrieve data from associated activity counters 78 and pass the information to DSP 14a to calculate a power value based on measured activities. The power management tasks, such as power compute task 84 and monitor task 82, can be executed along with other application tasks.

In the preferred embodiment, the power management tasks 38 and profiles 36 are implemented as JAVA class packages in a JAVA real-time environment.

The present invention provides significant advantages over the prior art. First, it provides for a fully dynamic power management. As the tasks executed in the distributed processing system 10 change, the power management can build new scenarios to ensure that thresholds are not exceeded. Further, as environmental conditions change, such as battery voltages dropping, the power management software can re-evaluate conditions and change scenarios, if necessary. For example, if the battery voltage (supply voltage) dropped to a point where Vdd could not be sustained at its nominal value, a lower frequency could be established, which would allow operation of the distributed processing system 10 at a lower Vdd. New scenarios could be built which would take the lower frequency into account. In some instances, more degradations would be introduced to compensate for the lower frequency. However, the lower frequency could provide for continued operation of the device, despite supply voltages that would normally be insufficient. Further, in situations where a lower frequency was acceptable, the device could operate at a lower Vdd (with the availability of a switched mode supply) in order to conserve power during periods of relatively low activity.

The power management software is transparent to the various tasks that it controls. Thus, even if a particular task does not provide for any power management, the power management software assumes responsibility for executing the task in a manner that is consistent with the power capabilities of the distributed processing system 10.

The overall operation of the power management software can be used with different hardware platforms, with different hardware and tasks accommodated by changing the profiles 36.

Although the Detailed Description of the invention has been directed to certain exemplary embodiments, various modifications of these embodiments, as well as alternative embodiments, will be suggested to those skilled in the art. The invention encompasses any modifications or alternative embodiments that fall within the scope of the Claims.

WHAT IS CLAIMED:

1. A method for controlling the execution of tasks in a processor comprising a plurality of processing modules, comprising the steps of:

calculating consumption information based on probabilistic values for activities associated with the tasks;

executing the tasks on said plurality of processing modules responsive to said consumption information

.

2. The method of claim 1 and further comprising the steps of:

monitoring actual activity occurrences in processing modules; and

modifying the execution of the tasks based on said monitoring step.

3. The method of claim 1 wherein said executing step comprises the step of executing the tasks on said plurality of processing modules responsive to said consumption information in order to provide the maximum performance within thermal constraints associated with the processing system.

4. The method of claim 1 wherein said executing step comprises the step of executing the tasks on said plurality of processing modules responsive to said consumption information in order to execute the tasks using the lowest possible energy consumption.

5. The method of claim 1 wherein said calculating step comprises the steps of:

generating a task allocation scenario;

estimating the activities for task allocation scenario;

computing the consumption associated with said activities.

6. The method of claim 5 wherein said step of generating a task allocation scenario comprises the step of receiving a task list describing the tasks to be executed and a task model describing the tasks.

7. The method of claim 6 wherein the task model includes initial estimates for each task.

8. The method of claim 7 wherein the task model further includes priority constraints associated with the tasks.

9. The method of claim 8 wherein said task model includes information regarding possible degradations associated with one or more of the tasks in said task list.

10. The method of claim 5 wherein said computing step comprises the step of computing the energy consumption associated with said activities.

11. The method of claim 5 wherein said computing step comprises the step of computing the power consumption associated with said activities.

12. A processing device comprising:

one or more processing modules for executing a plurality of tasks, said processing subsystems executing a power management function for:

calculating consumption information based on probabilistic values for activities associated with the tasks;

controlling the execution of the tasks on said processing modules responsive to said consumption information.

13. The processing device of claim 12 and further comprising counters for measuring activity occurrences and wherein said power management function further:

monitors said counters; and

modifies the execution of the tasks based on values in said counters.

14. The processing device of claim 12 wherein said power management function controls the execution of tasks on the processing modules responsive to said consumption information in order to provide the maximum performance within thermal constraints associated with the processing system.

15. The processing device of claim 12 wherein said power management function controls the execution of tasks on said processing modules responsive to said consumption information in order to execute the tasks using the lowest possible energy consumption.

16. The processing device of claim 12 wherein said power management function calculates the consumption information by:

generating a task allocation scenario;

estimating the activities for said task allocation scenario;

computing the consumption associated with said activities.

17. The processing device of claim 16 wherein said power management function generates a task allocation scenario by receiving a task list describing the tasks to be executed and a task model describing the tasks.

18. The processing device of claim 17 wherein the task model includes initial estimates for each task.

19. The processing device of claim 18 wherein the task model further includes priority constraints associated with the tasks.

20. The processing device of claim 19 wherein said task model includes information regarding possible degradations associated with one or more of the tasks in said task list.

21. The processing device of claim 16 wherein said power management function computes the consumption by computing the energy consumption associated with said activities.

22. The processing device of claim 16 wherein said power management function computes the consumption by computing the power consumption associated with said activities.

ABSTRACT OF THE DISCLOSURE

A distributed processing system (10) includes a plurality of processing modules, such as MPUs (12), DSPs (14), and coprocessors/DMA channels (16). Power management software (38) in conjunction with profiles (36) for the various processing modules and the tasks to executed are used to build scenarios which meet predetermined power objectives, such as providing maximum operation within package thermal constraints or using minimum energy. Actual activities associated with the tasks are monitored during operation to ensure compatibility with the objectives. The allocation of tasks may be changed dynamically to accommodate changes in environmental conditions and changes in the task list.

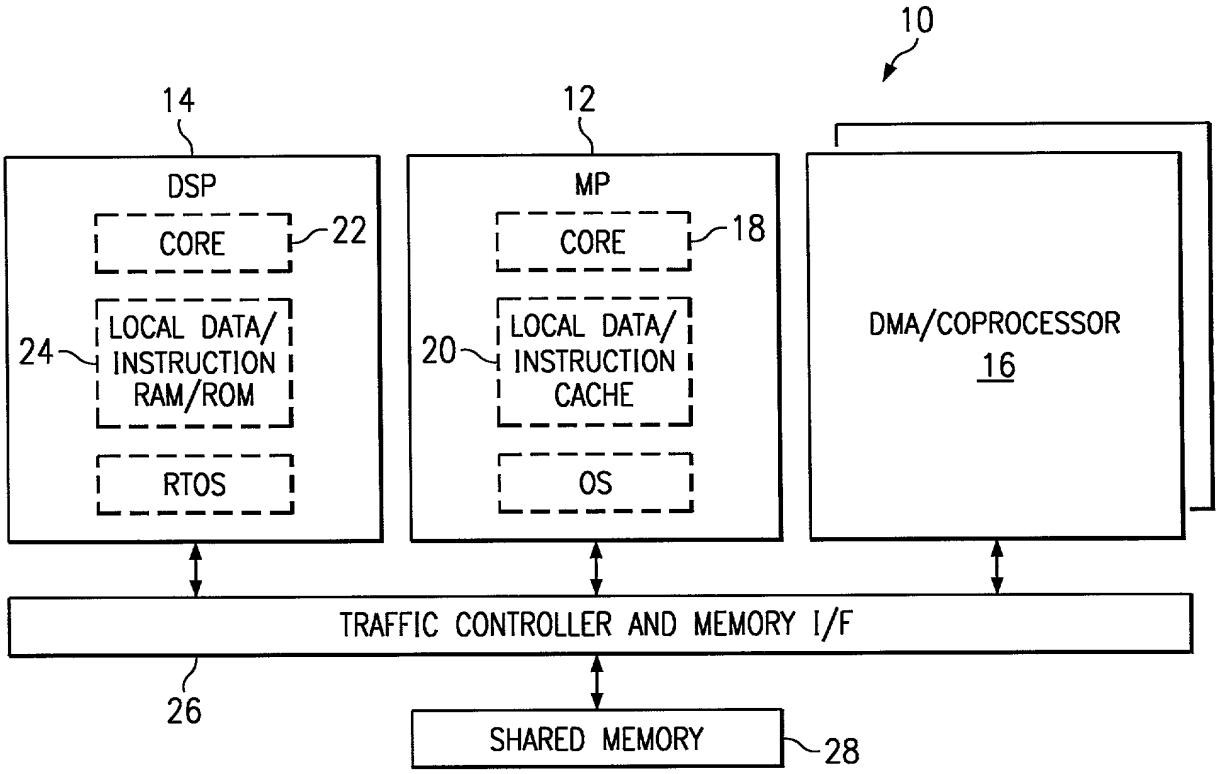
10
14
12
DSP
CORE
22
LOCAL DATA/
INSTRUCTION
RAM/ROM
24
RTOS
MP
CORE
18
LOCAL DATA/
INSTRUCTION
CACHE
20
OS
DMA/COPROCESSOR
16
TRAFFIC CONTROLLER AND MEMORY I/F
26
SHARED MEMORY
28

FIG. 1

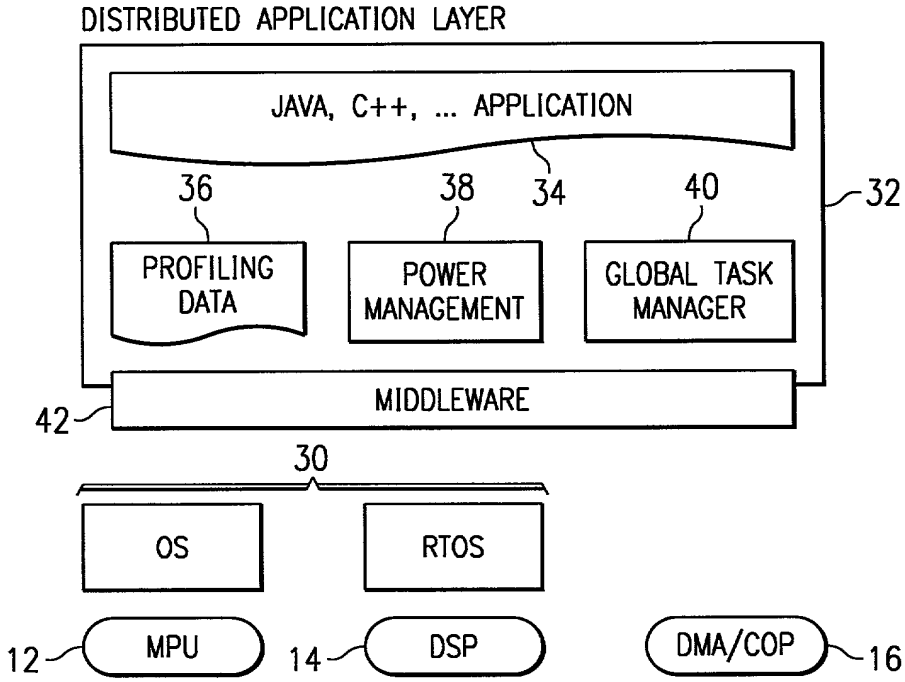
DISTRIBUTED APPLICATION LAYER
JAVA, C++, ... APPLICATION
34
36
38
40
32
PROFILING
DATA
POWER
MANAGEMENT
GLOBAL TASK
MANAGER
MIDDLEWARE
42
30
OS
RTOS
12
MPU
14
DSP
DMA/COP
16

FIG. 2

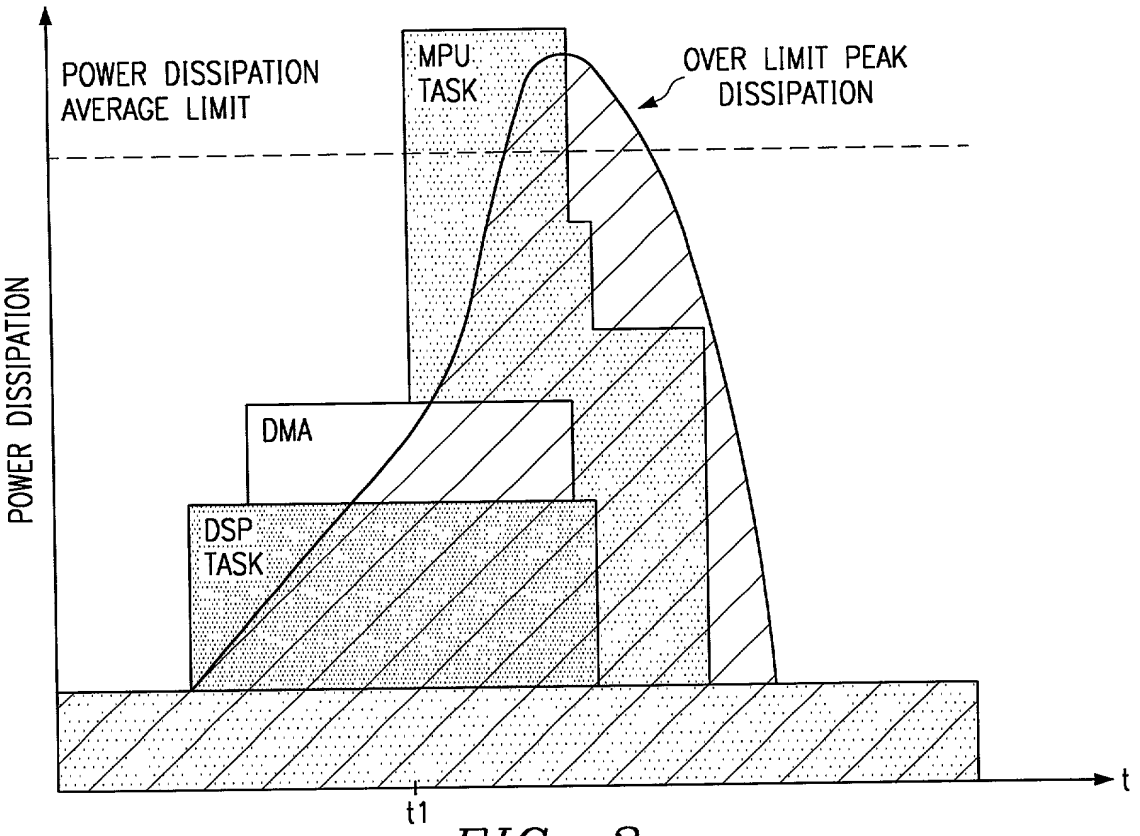
POWER DISSIPATION
AVERAGE LIMIT
MPU
TASK
OVER LIMIT PEAK
DISSIPATION
POWER DISSIPATION
DMA
DSP
TASK
t
t1

FIG. 3a

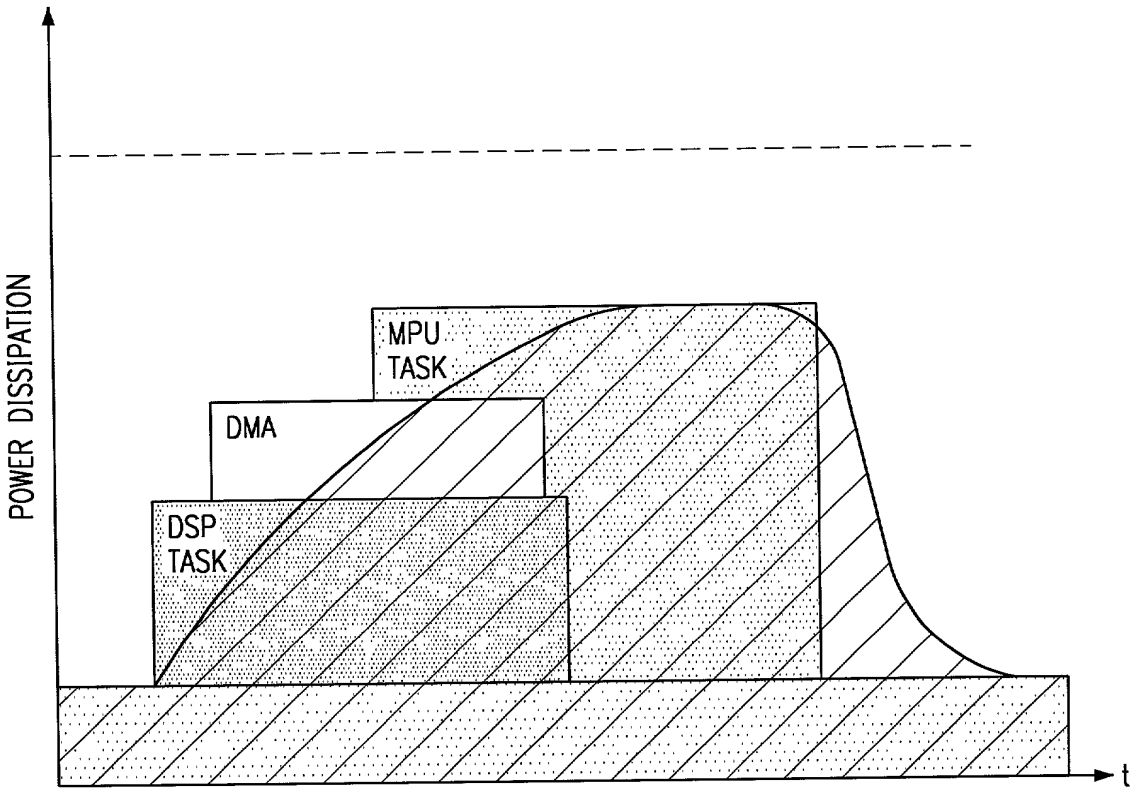
POWER DISSIPATION
MPU
TASK
DMA
DSP
TASK
t

FIG. 3b

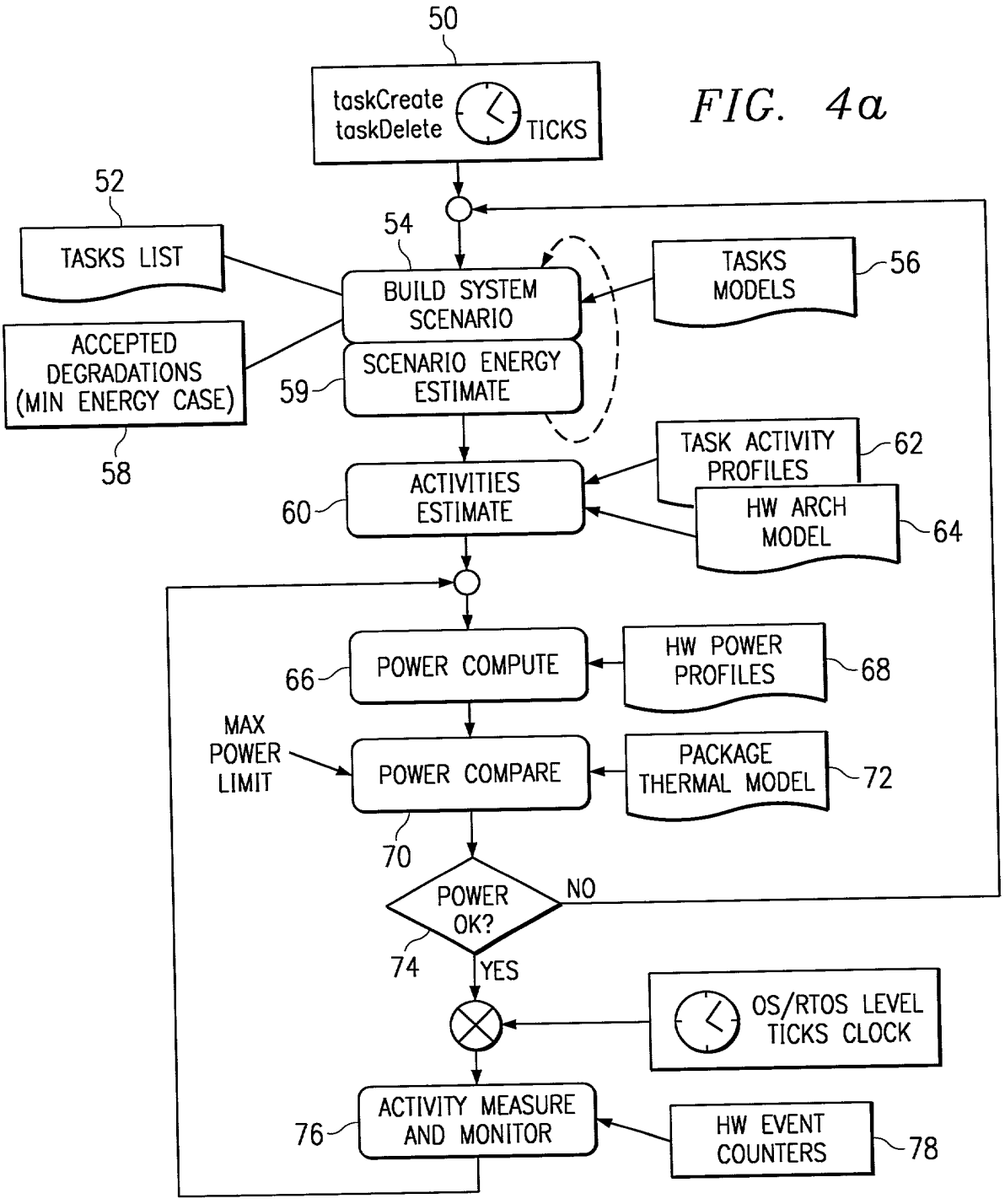
50
taskCreate
taskDelete
TICKS
52
TASKS LIST
54
BUILD SYSTEM SCENARIO
TASKS MODELS
56
ACCEPTED DEGRADATIONS (MIN ENERGY CASE)
58
59
SCENARIO ENERGY ESTIMATE
TASK ACTIVITY PROFILES
62
60
ACTIVITIES ESTIMATE
HW ARCH MODEL
64
66
POWER COMPUTE
HW POWER PROFILES
68
MAX POWER LIMIT
POWER COMPARE
PACKAGE THERMAL MODEL
72
70
POWER OK?
NO
74
YES
OS/RTOS LEVEL TICKS CLOCK
76
ACTIVITY MEASURE AND MONITOR
HW EVENT COUNTERS
78

FIG. 4a

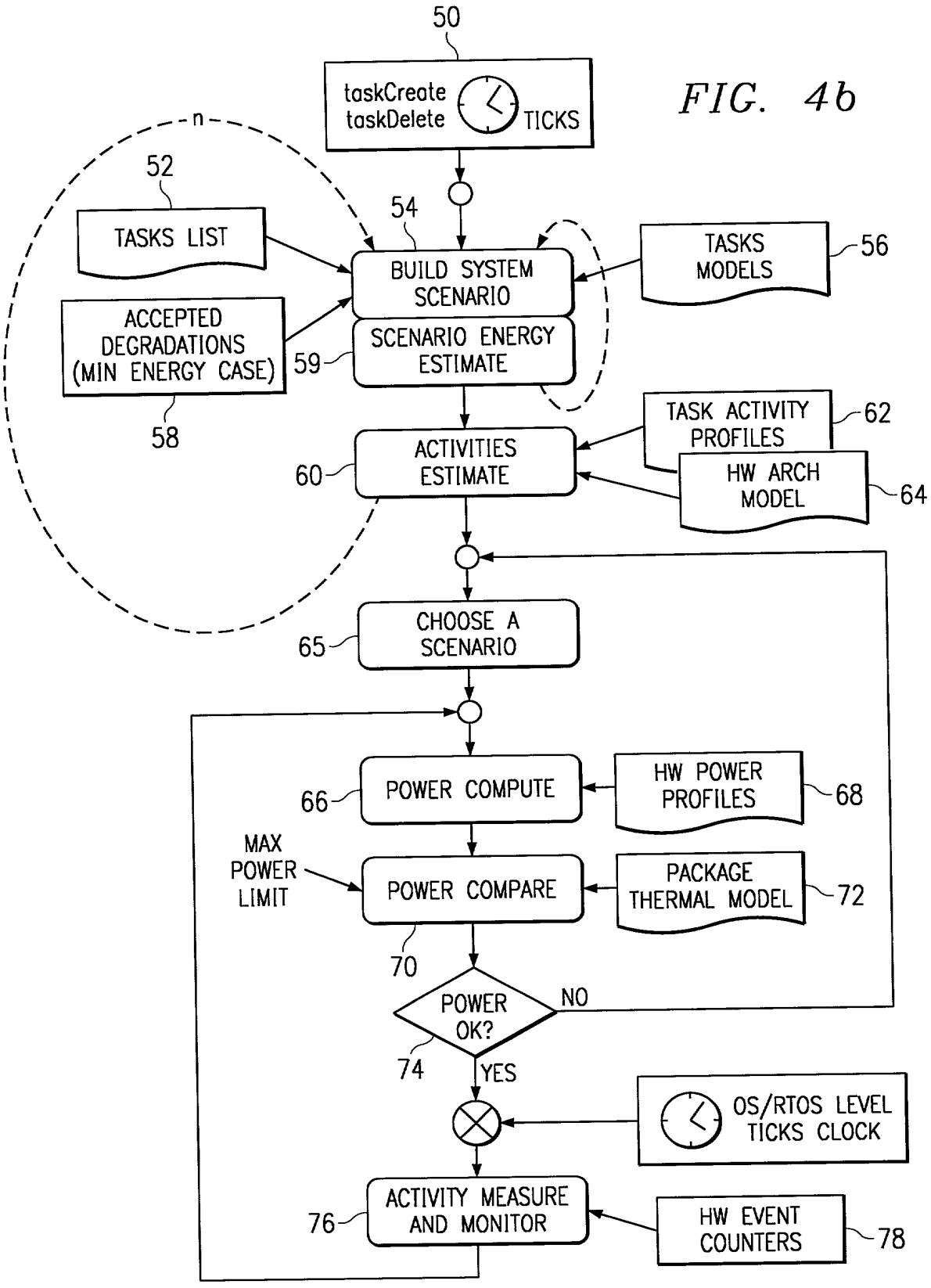
50
taskCreate
taskDelete
TICKS
n
52
TASKS LIST
54
BUILD SYSTEM SCENARIO
TASKS MODELS
56
ACCEPTED DEGRADATIONS (MIN ENERGY CASE)
58
59
SCENARIO ENERGY ESTIMATE
TASK ACTIVITY PROFILES
62
60
ACTIVITIES ESTIMATE
HW ARCH MODEL
64
65
CHOOSE A SCENARIO
66
POWER COMPUTE
HW POWER PROFILES
68
MAX POWER LIMIT
POWER COMPARE
PACKAGE THERMAL MODEL
72
70
POWER OK?
NO
74
YES
OS/RTOS LEVEL TICKS CLOCK
76
ACTIVITY MEASURE AND MONITOR
HW EVENT COUNTERS
78

FIG. 4b

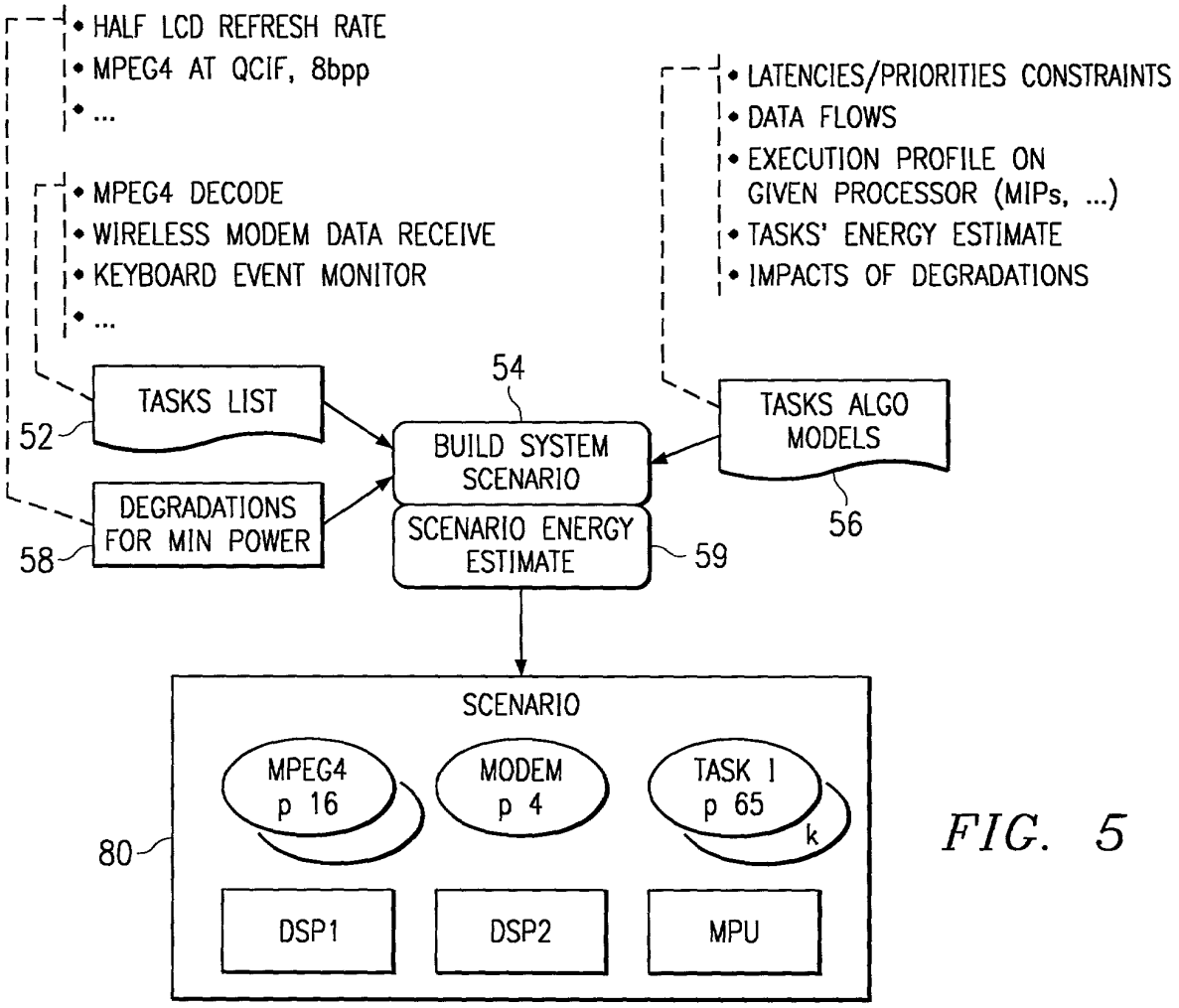
• HALF LCD REFRESH RATE
• MPEG4 AT QCIF, 8bpp
• ...
• MPEG4 DECODE
• WIRELESS MODEM DATA RECEIVE
• KEYBOARD EVENT MONITOR
• ...
• LATENCIES/PRIORITIES CONSTRAINTS
• DATA FLOWS
• EXECUTION PROFILE ON GIVEN PROCESSOR (MIPs, ...)
• TASKS' ENERGY ESTIMATE
• IMPACTS OF DEGRADATIONS
TASKS LIST
52
DEGRADATIONS FOR MIN POWER
58
54
BUILD SYSTEM SCENARIO
SCENARIO ENERGY ESTIMATE
59
TASKS ALGO MODELS
56
SCENARIO
MPEG4 p 16
MODEM p 4
TASK 1 p 65
k
80
DSP1
DSP2
MPU

FIG. 5

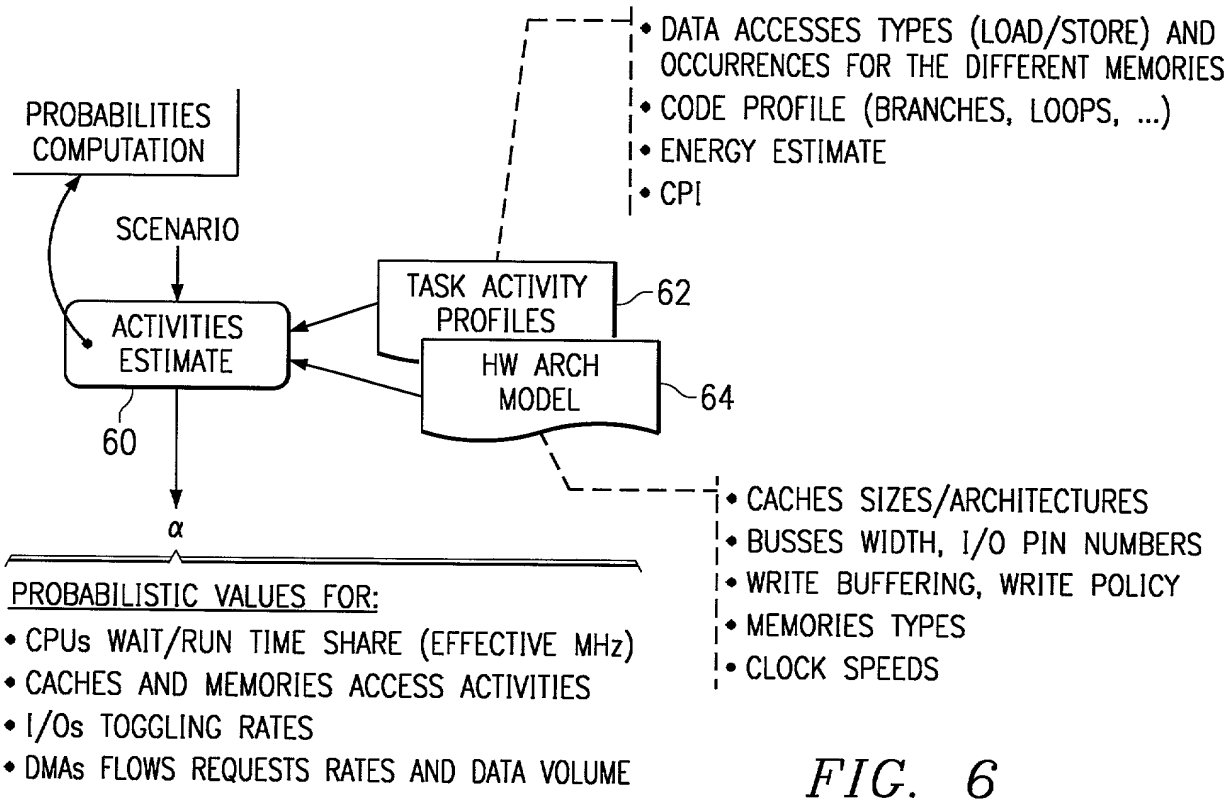
PROBABILITIES COMPUTATION
SCENARIO
ACTIVITIES ESTIMATE
60
α
TASK ACTIVITY PROFILES
62
HW ARCH MODEL
64
• DATA ACCESSES TYPES (LOAD/STORE) AND OCCURRENCES FOR THE DIFFERENT MEMORIES
• CODE PROFILE (BRANCHES, LOOPS, ...)
• ENERGY ESTIMATE
• CPI
• CACHES SIZES/ARCHITECTURES
• BUSSES WIDTH, I/O PIN NUMBERS
• WRITE BUFFERING, WRITE POLICY
• MEMORIES TYPES
• CLOCK SPEEDS
PROBABILISTIC VALUES FOR:
• CPUs WAIT/RUN TIME SHARE (EFFECTIVE MHz)
• CACHES AND MEMORIES ACCESS ACTIVITIES
• I/Os TOGGLING RATES
• DMAs FLOWS REQUESTS RATES AND DATA VOLUME

FIG. 6

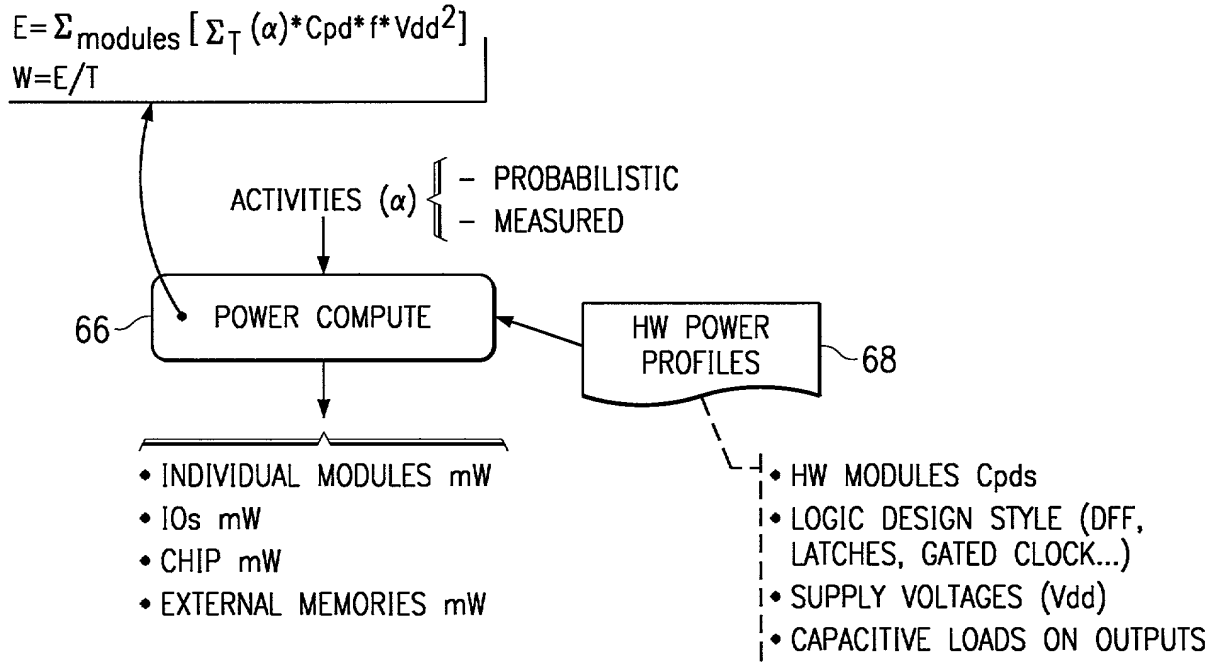
E=Σmodules [ΣT (α)*Cpd*f*Vdd2]
W=E/T
ACTIVITIES (α)
– PROBABILISTIC
– MEASURED
66
POWER COMPUTE
HW POWER PROFILES
68
• INDIVIDUAL MODULES mW
• IOs mW
• CHIP mW
• EXTERNAL MEMORIES mW
• HW MODULES Cpds
• LOGIC DESIGN STYLE (DFF, LATCHES, GATED CLOCK...)
• SUPPLY VOLTAGES (Vdd)
• CAPACITIVE LOADS ON OUTPUTS

FIG. 7

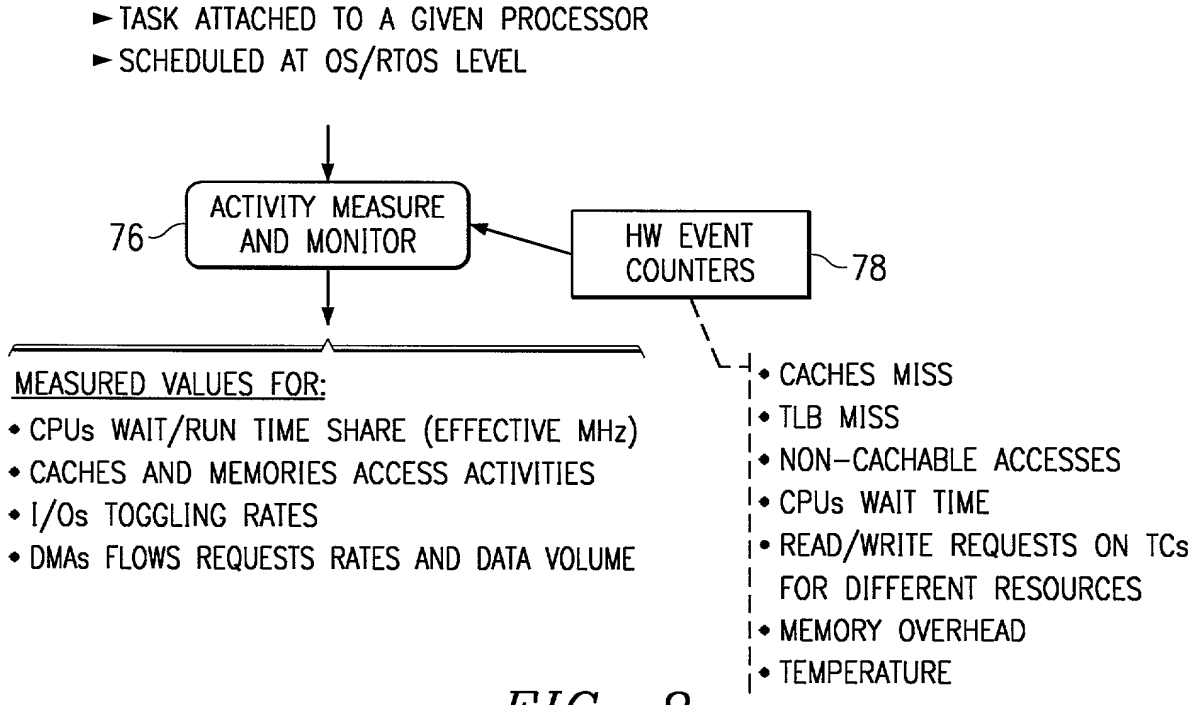
►TASK ATTACHED TO A GIVEN PROCESSOR
►SCHEDULED AT OS/RTOS LEVEL
76
ACTIVITY MEASURE AND MONITOR
HW EVENT COUNTERS
78
MEASURED VALUES FOR:
• CPUs WAIT/RUN TIME SHARE (EFFECTIVE MHz)
• CACHES AND MEMORIES ACCESS ACTIVITIES
• I/Os TOGGLING RATES
• DMAs FLOWS REQUESTS RATES AND DATA VOLUME
• CACHES MISS
• TLB MISS
• NON-CACHABLE ACCESSES
• CPUs WAIT TIME
• READ/WRITE REQUESTS ON TCs FOR DIFFERENT RESOURCES
• MEMORY OVERHEAD
• TEMPERATURE

FIG. 8

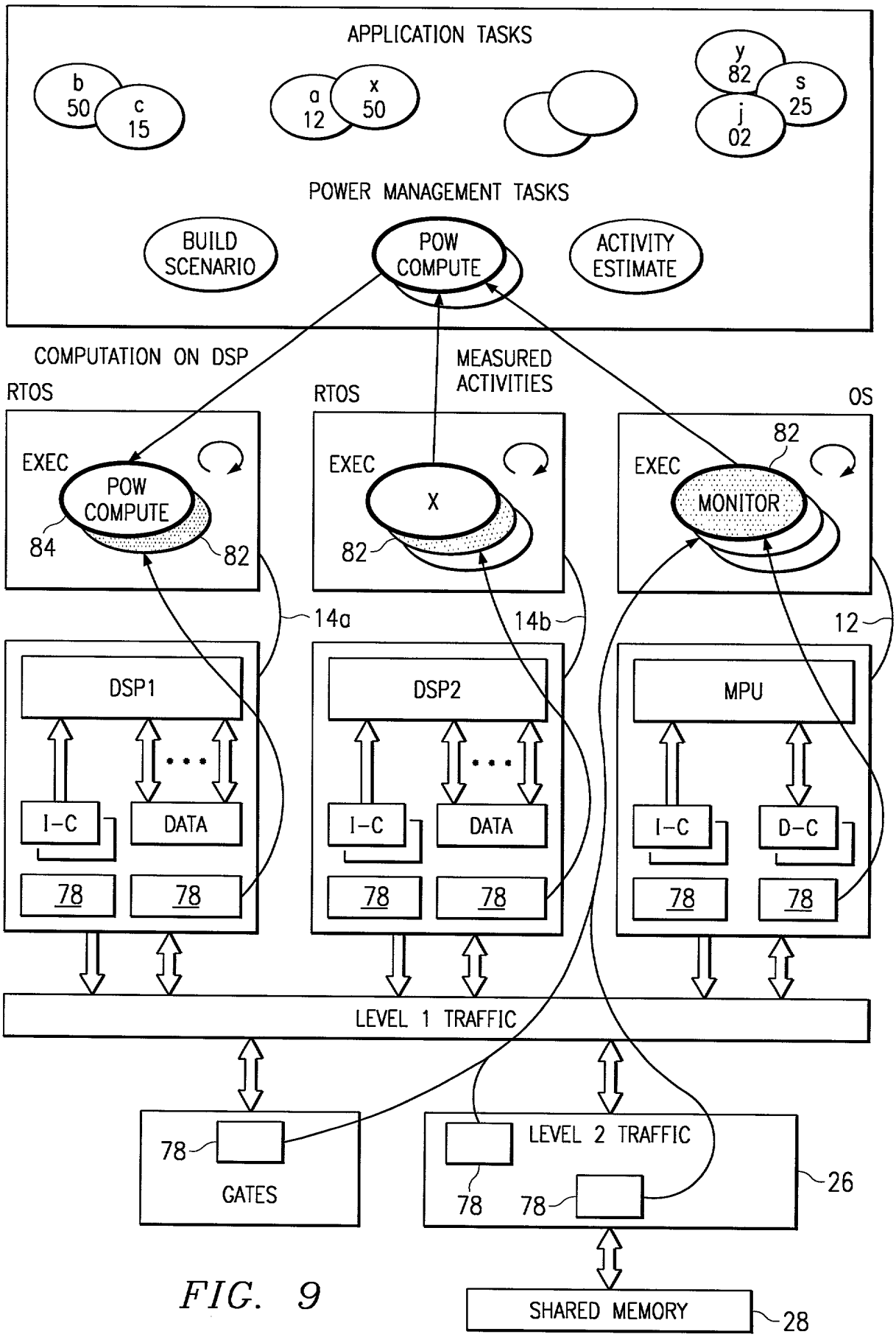
APPLICATION TASKS
b
50
c
15
a
12
x
50
y
82
s
25
j
02
POWER MANAGEMENT TASKS
BUILD SCENARIO
POW COMPUTE
ACTIVITY ESTIMATE
COMPUTATION ON DSP
MEASURED ACTIVITIES
RTOS
RTOS
OS
EXEC
POW COMPUTE
84
82
EXEC
X
82
EXEC
82
MONITOR
14a
14b
12
DSP1
DSP2
MPU
I-C
DATA
I-C
DATA
I-C
D-C
78
78
78
78
78
78
LEVEL 1 TRAFFIC
78
GATES
LEVEL 2 TRAFFIC
78
78
26
SHARED MEMORY
28

FIG. 9